融合协同过滤的线性回归推荐算法 *

庞海龙，赵　辉†，李万龙，马　莹，崔　岩

(长春工业大学 计算机科学与工程学院, 长春 130012)

摘　要：针对传统协同过滤算法中存在数据稀疏问题，提出融合协同过滤的线性回归推荐算法。根据用户对项目的评分以及用户和项目自身特征，构建用户间和项目间相似矩阵。基于相似矩阵，选出用户和项目最近邻集合，分别通过基于用户和基于项目的协同过滤算法来预测用户已评分项目的评分，将预测评分与真实评分的差值作为特征，组合在一起生成新的训练数据。把新的训练数据作为线性回归模型的输入，根据训练好的模型预测未知评分，采用 Top-N 算法产生推荐列表。在 MovieLens 数据集上进行实验。实验结果表明，新算法的推荐准确性较传统协同过滤算法有显著提高。

关键词：线性回归；协同过滤；相似性；推荐算法

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.11.0732

Linear regression recommendation algorithm with collaborative filtering

Pang Hailong, Zhao Hui†, Li Wanlong, Ma Ying, Cui Yan

(*College of Computer Science & Engineering*, *Changchun University of Technology*, *Changchun* 130012, *China*)

Abstract: This paper proposed a linear regression algorithm to integrate collaborative filtering based on the data sparse influence of the traditional collaborative filtering algorithm. Firstly, it built a similarity matrix between the user and the project based on the user's rating of the project, as well as the user and the project's own characteristics. Secondly, based on the similarity matrix, it selected the user and project nearest neighbor set. It predicted the score that the users had graded respectively by the way of collaborative filtering algorithms based on the user and the project. And it would take the difference between predicted scores and the real scores as features to generate new training data, and regard the new training data as the input of the linear regression model. Finally, according to the training model, it could predict the unknown score , and used the Top-N algorithm to generate the recommended list. It conducted the experiment on the MovieLens data set. The experimental result shows that the proposed accuracy of the new algorithm improves compared with the traditional collaborative filtering algorithm.

Key words: linear regression; collaborative filtering; similarity; recommendation algorithm

0　引言

随着互联网技术的飞速发展，各类信息瞬间暴增，导致严重的“信息过载”问题[1]。一方面，从用户的角度来看，从海量的数据中获取自己感兴趣的信息变的越来越困难；另一方面，从服务提供商的角度来看，用户能够提供的有效信息少之又少，为他们提供个性化的需求变得愈加困难。推荐系统[2]作为一种信息过滤技术，在解决上述问题中起到了举足轻重的作用。它根据用户偏好向用户推荐其可能感兴趣的项目(如音乐、电影、图书等)。

目前推荐系统应用最广泛的技术之一是协同过滤算法[3,4]，其主要分为基于用户的协同过滤算法和基于项目的协同过滤算法。基于用户的协同过滤算法给用户推荐与其兴趣相似的其他用户感兴趣的项目；基于项目的协同过滤算法给用户推荐与其之前感兴趣的项目相似的项目[5]。这两种传统协同过滤算法都是先根据用户对项目的评分计算用户或项目之间的相似性，然后找出用户或项目的最近邻集合，最后根据 Top-N 算法产生推荐列表进行推荐。然而随着推荐系统中用户和项目的数量不断扩增，传统协同过滤算法面临着扩展性、数据稀疏等问题。其中，数据扩展性指随着数据量的增加，无法及时计算出相似用户或项目，导致推荐延误；数据稀疏性指用户一般只对很少的项目进行评分，数据量越大，评分信息显得越少，相似性计算不够准确，导致推荐准确度降低。

为了解决上述问题，文献[6~8]通过矩阵分解降低维数，减

收稿日期：2017-11-01；**修回日期：**2018-01-01　　**基金项目：**国家自然科学基金资助项目(61472049)；吉林省教育厅“十二五”科学技术研究项目(2014132)

作者简介：庞海龙（1987-），男，吉林长春人，硕士研究生，主要研究方向为推荐系统、智能计算；赵辉（1972-），女（通信作者），教授，博士，主要研究方向为智能计算、搜索引擎（zhaohui@mail.ccut.edu.cn）；李万龙（1963-），男，教授，博士，主要研究方向为软件工程、智能系统；马莹（1993-），女，硕士研究生，主要研究方向为自然语言处理、智能计算；崔岩（1993-），男，硕士研究生，主要研究方向为推荐系统、智能计算.

少存储空间，降低计算复杂度,以解决数据扩展性问题。但是分解算法不仅丢失原始评分信息,而且还容易产生过拟合的现象。文献[9~11]采用聚类技术，对用户或项目进行聚类，缩小相似性范围填充评分值，一定程度上缓解了数据稀疏性问题；但是忽略了用户兴趣的差异性，推荐精度难以保证。文献[16]通过统计用户或物品的评分频次建立线性回归模型，进而利用该模型对未知评分直接根据历史评分频次进行预测。本文提出一种融合协同过滤的线性回归推荐算法(linear regression recommendation algorithm with collaborative filtering，LRCF)。首先，将用户历史评分以及用户和项目自身特征融入到相似性计算中，根据相似性矩阵选出最近邻集合；其次，基于协同过滤算法预测用户已评分项目的评分，通过预测评分与真实评分的差值建立线性回归模型；最后，根据该模型预测未知评分，从整体上提高用户预测评分的准确性。

1 相关工作

1.1 协同过滤算法

协同过滤算法的中心思想是在整个空间寻找用户或项目的前 k 个最近邻，核心是计算相似性。相似性计算的常用方法有余弦相似性[12]、修正的余弦相似性[13]、皮尔逊相关系数[14]等。以基于项目的协同过滤算法为例，分别介绍三种相似性的计算公式、预测评分公式以及 top-N 算法推荐。

余弦相似性计算公式为

$$sim(i,j)=\frac{\sum_{u\in U_{i,j}}R_{u,i}\bullet R_{u,j}}{\sqrt{\sum_{u\in U_{i,j}}R_{u,i}^2}\sqrt{\sum_{u\in U_{i,j}}R_{u,j}^2}} \tag{1}$$

其中：i、j 分别表示整个项目空间的两个项目；$\boldsymbol{U}_{i,j}$ 表示对项目 i、j 评分过的用户集合；$\boldsymbol{R}_{u,i}$ 表示用户 u 对项目 i 的评分；$\boldsymbol{R}_{u,j}$ 表示用户 u 对项目 j 的评分。

修正的余弦相似性计算公式为

$$sim(i,j)=\frac{\sum_{u\in U_{i,j}}(R_{u,i}-\overline{R_u})(R_{u,j}-\overline{R_u})}{\sqrt{\sum_{u\in U_{i,j}}(R_{u,i}-\overline{R_u})^2}\sqrt{\sum_{u\in U_{i,j}}(R_{u,j}-\overline{R_u})^2}} \tag{2}$$

其中：$\overline{R_u}$ 表示用户 u 对已评分项目的平均评分。

皮尔逊相关系数计算公式为

$$sim(i,j)=\frac{\sum_{u\in U_{i,j}}(R_{u,i}-\overline{R_i})(R_{u,j}-\overline{R_j})}{\sqrt{\sum_{u\in U_{i,j}}(R_{u,i}-\overline{R_i})^2}\sqrt{\sum_{u\in U_{i,j}}(R_{u,j}-\overline{R_j})^2}} \tag{3}$$

其中：$\boldsymbol{U}_{i,j}$ 表示对项目 i 和 j 共同评分过的所有用户集合；$\overline{R_i}$ 表示对项目 i 的平均评分；$\overline{R_j}$ 表示对项目 j 的平均评分。

预测评分公式为

$$p_{u,i}=\overline{R_i}+\frac{\sum_{i_k\in I_n}(R_{u,i_k}-\overline{R_{i_k}})sim(i,i_k)}{\sum_{i_k\in I_n}|sim(i,i_k)|} \tag{4}$$

在所有项目中寻找与目标项目 i 相似性最高的前 k 个项目构成项目 i 的最近邻 $I_n=\{i_1,i_2,\ldots,i_k\}$。在最近邻 I_n 确定以后，预测用户 u 对未评分项目 i 的评分，如式(4)所示。

根据式(4)计算出所有未对项目 i 评分的用户的预测评分，记为一个集合 $\boldsymbol{C}$，对集合 $\boldsymbol{C}$ 按降序的方式排序，最后把排序靠前的 N 个项目推荐给用户。

1.2 线性回归算法

线性回归[15,16]是根据给定的一系列特征，对给定特征和实际值之间的组合关系进行分析，并通过线性组合的方式来拟合真实值。模型表示形式如式（5）所示。

$$h_\theta(x)=\sum_{i=1}^{m}\theta_i x_i=\theta^T x \tag{5}$$

其中：m 表示特征个数；$h_\theta(x)$ 表示预测值；θ^T 表示参数向量；x 表示特征向量。预测值与真实值之间存在一定的误差，这个误差服从高斯分布，最终误差损失函数表示形式如下：

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2 \tag{6}$$

其中：$J(\theta)$ 表示误差平方和；$y^{(i)}$ 表示真实值。利用梯度下降法优化求解 $J(\theta)$，求 $J(\theta)$ 对参数 θ 的偏导，然后利用式(7)迭代更新参数 θ,直到达到最大迭代次数，求得参数 θ。

$$\theta_{j+1}^{(i)}=\theta_j^{(i)}-\alpha\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)} \tag{7}$$

其中：$\theta_j^{(i)}$ 表示特征向量 $x^{(i)}$ 的第 j 个参数；α 表示学习率；$x_j^{(i)}$ 表示特征向量 $x^{(i)}$ 的第 j 个值。

2 融合协同过滤的线性回归推荐算法构建

2.1 问题定义

为了更好地描述本文提出的算法，现对用户集合、项目集合、用户特征矩阵、项目特征矩阵以及用户—项目评分矩阵进行符号化定义。

定义 1 $\boldsymbol{U}$ 表示用户集合，形式如下：

$$U=\{u_1,u_2,\ldots,u_m\}$$

其中：u_m 表示第 m 个用户。

定义 2 $\boldsymbol{I}$ 表示项目集合，形式如下：

$$I=\{i_1,i_2,\ldots,i_n\}$$

其中：i_n 表示第 n 个项目。

定义 3 ***UFeature*** 表示用户特征矩阵，形式如下：

用户	f_1	f_2	f_3	…	f_p
u_1	$uf_{1,1}$	$uf_{1,2}$	$uf_{1,3}$	…	$uf_{1,p}$
u_2	$uf_{2,1}$	$uf_{2,2}$	$uf_{2,3}$	…	$uf_{2,p}$
…	…	…	…	…	…
u_m	$uf_{m,1}$	$uf_{m,2}$	$uf_{m,3}$	…	$uf_{m,p}$

其中：$uf_{m,P}$ 表示用户 u_m 的第 p 个特征。

定义 4 ***IFeature*** 表示项目特征矩阵，形式如下：

项目	f_1	f_2	f_3	…	f_p
i_1	$if_{1,1}$	$if_{1,2}$	$if_{1,3}$	…	$if_{1,q}$
i_2	$if_{2,1}$	$if_{2,2}$	$if_{2,3}$	…	$if_{2,q}$
…	…	…	…	…	…
i_n	$if_{n,1}$	$if_{n,2}$	$if_{n,3}$	…	$if_{n,q}$

其中：*ifn,q* 表示项目 i_n 的第 *q* 个特征。

定义 5 *R* 表示用户-项目评分矩阵，形式如下：

用户	i_1	i_2	i_3	...	i_n
u_1	$R_{1,1}$	$R_{1,2}$	$R_{1,3}$	...	$R_{1,n}$
u_2	$R_{2,1}$	$R_{2,2}$	$R_{2,3}$	...	$R_{2,n}$
...	...	...	...	...	...
u_m	$R_{m,1}$	$R_{m,2}$	$R_{m,3}$	...	$R_{m,n}$

其中：$R_{m,n}$ 表示用户 u_m 对项目 i_n 的评分值。

2.2 算法构建

算法构建分为生成训练数据集和产生推荐列表两个阶段。

阶段 1：生成训练数据集，流程如图 1 所示。输入、输出、算法步骤如下：

输入:预处理后的用户特征矩阵 ***UFeature***,预处理后的项目特征矩阵 ***IFeature***，用户—项目评分矩阵 ***R***，用户集合 ***U***，项目集合 ***I***，最近邻数 ***k***。

输出:训练数据。

算法步骤：

a) 在 ***R*** 中找出用户 *u* 已评分的项目集合 $I_u=\{i\,|\,i\in I,R_{u,i}\neq\varnothing\}$ 和对项目 *i* 评分过的用户集合 $U_i=\{u\,|\,u\in U,R_{u,i}\neq\varnothing\}$。

b) 根据 I_u、U_i 分别生成两两项目对集合 $ipairs=\{<i_a,i_b>|\,i_a,i_b\in I_u\}$ 和用户对集合 $upairs=\{<u_a,u_b>|\,u_a,u_b\in U_i\}$。

c)对于 *ipairs* 中的每一对 $<i_a,i_b>$ 在 **IFeature** 中找出 i_a 和 i_b 对应的行，利用式(2)计算相似性 $sim(i_a,i_b)$;同样的方式利用式(3)计算 *upairs* 中每一对 $<u_a,u_b>$ 的相似性 $sim(u_a,u_b)$。

d)循环执行 a)b)c),得到每个用户和每个项目的相似性，分别构建用户相似性矩阵 $usim(m,m)$ 和项目相似性矩阵 $isim(n,n)$。

e)在 ***R*** 上计算每一个用户在 *usim* 的相似性，取相似性最高的前 *k* 个用户构成用户最近邻集合 $N_u=\{\{u_1,u_2,...,u_k\}\,|\,u\in usim\}$ 并保存；同理，计算每一个项目在 *isim* 的相似性，取相似性最高的前 *k* 个项目构成项目最近邻集合 $N_i=\{\{i_1,i_2,...,i_k\}\,|\,i\in isim\}$ 并保存。

f)遍历 ***R***，选择项目 *i* 对应的最近邻集合 N_i，根据式(4)计算用户 *u* 对项目 *i* 的预测评分 $p^i{}_{u,i}$,将 $p^i{}_{u,i}$ 与 $R_{u,i}$ 的差值记为 x_1^i；同理，选择用户 *u* 对应的最近邻集合 N_u，类比式(4)计算用户 u 对项目 i 预测评分 $p^u{}_{u,i}$，将 $p^u{}_{u,i}$ 与 $R_{u,i}$ 的差值记为 x_2^u；最后将 x_1^i、x_2^u、$R_{u,i}$ 组成特征，构造新的数据集 $data=\{\{x_2^u,x_1^i,R_{u,i}\}\,|\,i\in I,u\in U,R_{u,i}\in R\}$ 并保存。

阶段 2：产生推荐列表，流程如图 2 所示。输入、输出、算法步骤如下。

输入：训练数据，目标用户 *u*。

输出：目标用户 *u* 的推荐列表。

算法步骤：

a)将训练数据的前两列作为线性回归模型的输入参数 $X=\{x_1,x_2\}$，最后一列作为样本标签 $Y=\{y_{u,i}\}$。

b)根据式(5)(6)建立线性回归模型，利用梯度下降法优化求解下边损失函数 $J(\theta)$，$J(\theta)=\frac{1}{2}\sum_{j=1}^{2}(\theta^T x_j-y_{u,i})^2$。

c)求损失函数 $J(\theta)$ 对参数 θ 的偏导，然后利用式(7)迭代更新参数 θ，直到达到最大迭代次数，将求得的参数 θ，带入式(5)，得到 **LRCF** 模型。

d)根据 **LRCF** 预测目标用户 *u* 对未评分项目的评分，使用 top-N 算法生成推荐列表。

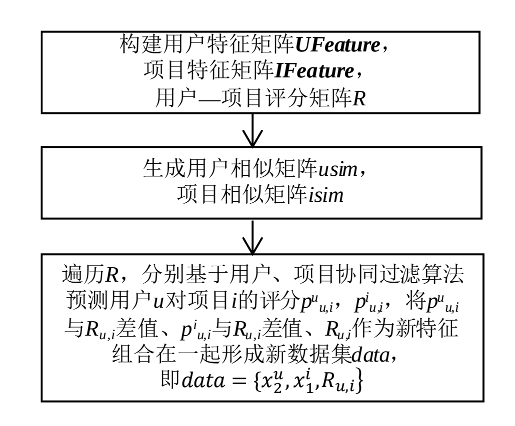


图 1 生成训练数据流程

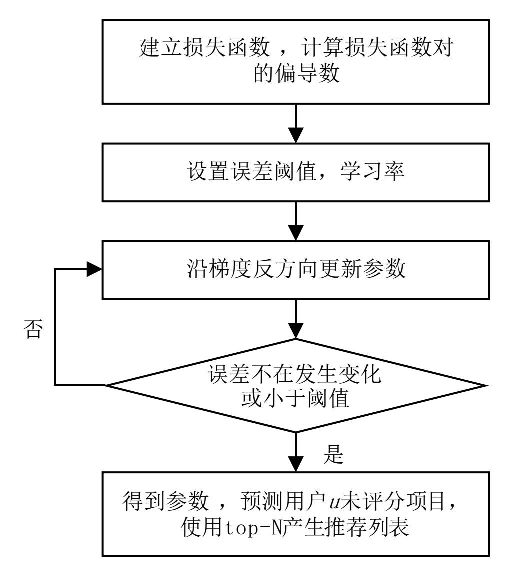


图 2 产生推荐列表流程

3 实验分析

3.1 实验数据

本文采用的实验数据是由 GroupLens 提供的 MovieLens100k 数据集[17]。该数据集提供了用户特征数据集、电影特征数据集、用户评分数据集等。其中，用户特征数据集包含 943 条记录，每一条记录了用户 id、年龄、性别、职业、邮编，如表 1 所示；电影特征数据集包含 1 682 条记录，每一条记录了电影 id、电影名、上映日期，IMDb 上的网址，类别，如表 2 所示；评分数据集包含 100 000 条评分记录，记录了 943

个用户对 1 682 部电影的评分，每个用户至少对 20 部电影进行了评分，评分区间是 1~5，数值大小代表用户对电影的喜爱程度。评分数据集的稀疏度是 1-100000/943*1682 = 0.93695。实验数据集的各项基本特征如表 3 所示。实验中将数据集按照一定比例划分为训练集和测试集，其中 80%作为训练集，20%作为测试集。

表 1 用户特征示例

用户 id	1	2
年龄	24	53
性别	M	F
职业	technician	other
邮编	85711	94043

表 2 项目属性示例

电影 id	1
电影名	Toy Story
上映日期	01-Jan-1995
IMDb 上的网址	http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
类别	Animation|Children's|Comedy

表 3 实验数据统计信息

用户数目(个)	943
项目数目(个)	1682
评分记录数(条)	100000
用户最大评分数(条)	685
用户最小评分数(条)	20
用户平均评分(分)	3.52986
稀疏度(%)	93.695%

3.2 度量标准

本实验采用目前最常用的一种推荐质量度量标准，即均方根误差(root mean square error，RMSE)。RMSE 的值越小，推荐质量越好。RMSE 定义如下：

$$RMSE = \sqrt{\frac{\sum_{u,i \in R_{Test}} \left(R_{u,i} - p_{u,i}\right)^2}{N}} \quad (8)$$

其中：N 表示测试集评分数据个数；R_{test} 测试集评分数据集合；$R_{u,i}$ 用户 u 对项目 i 的真实打分；$p_{u,i}$ 用户 u 对项目 i 的预测打分。

3.3 实验结果及分析

考虑各个参数对本文算法的影响，本节先通过部分实验找到最优参数的设定，在最优参数确定的基础上对本文算法和传统协同过滤算法进行比较。本实验的用户相似性度量方法是皮尔逊相关系数，项目相似性度量方法是修正的余弦相似性和皮尔逊相关系数。

1）最近邻数分析

考察用户和项目最近邻数对本文算法(LRCF)推荐精度的影响。将最近邻数在 5~305 间变动，固定学习率 α=0.01。如图 2 所示，随着用户最近邻数 k 值的增加，**LRCF** 的 RMSE 值减小，推荐准确率上升，用户最近邻数 k 取 275 时，**LRCF** 的 RMSE 值达到最小，随后再增加 k 值，**LRCF** 的 RMSE 值不在发生变化，推荐准确率保持不变；如图 3 所示，k 值在 5~25 间，**LRCF** 的 RMSE 的值一直在减小，k 值在 25~55 间，RMSE 的值一直在增大，随后再增加 k 值，**LRCF** 的 RMSE 的值一直在减小，推荐准确率上升，项目最近邻数 k 取 275 时，**LRCF** 的 RMSE 值达到最小，随后再增加 k 值，**LRCF** 的 RMSE 值不在发生变化，推荐准确率保持不变。

2）学习率 α 分析

考察不同学习率 α 对本文算法的影响。将学习率 α 在 0.001~1 间变动。如图 4 所示，学习率 α=0.1 时，本文算法的 RMSE 值到达最小，随后再增加 α 值，本文算法的 RMSE 值不在发生变化。

因此，针对于 **LRCF** 分别设置用户最近邻 k=275、项目最近邻 k=275、学习率 α=0.1 进行后续实验。

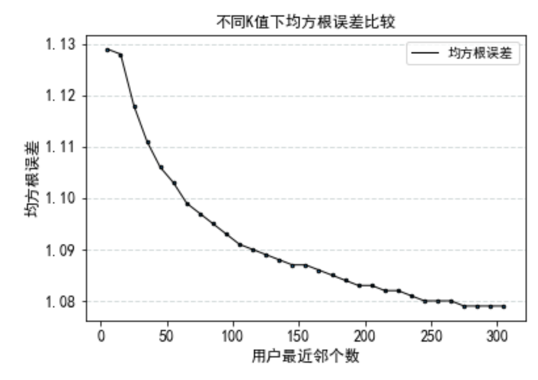


图 2 用户最近邻数 k 对于均方根误差的影响

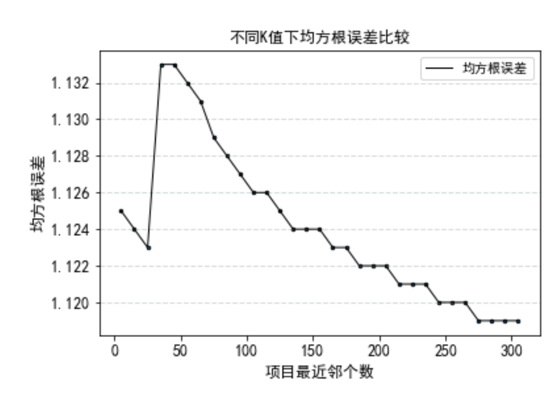


图 3 项目最近邻数 k 对于均方根误差的影响

3）与传统协同过滤算法比较

将本文算法与 **userCF**、**itemCF** 作比较。如图 5 所示，本文提出的融合协同过滤的线性回归推荐算法具有最小的 RMSE，由此可知本文提出的推荐算法准确性较传统协同过滤算法有显著提高。

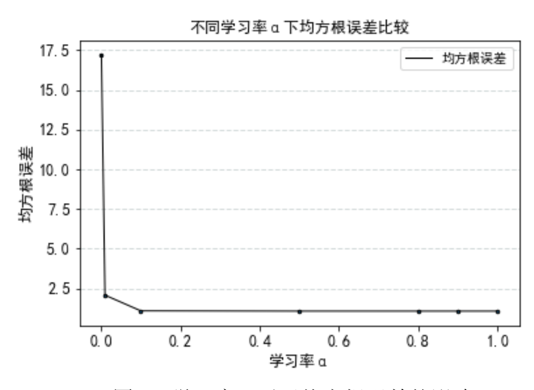


图 4 学习率α对于均方根误差的影响

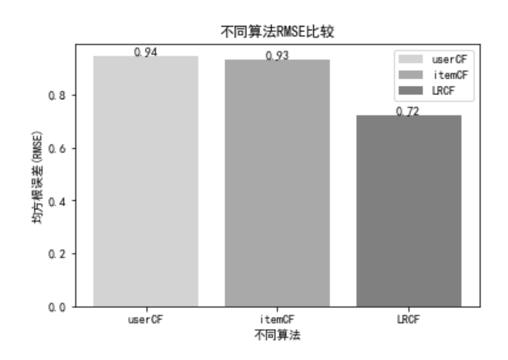


图 5 不同算法对于均方根误差的影响

4 结束语

本文提出融合协同过滤的线性回归推荐算法，通过用户—项目评分矩阵以及用户特征和项目属性构建相似矩阵，准确计算用户和项目的最近邻集合，有效克服了因数据稀疏导致推荐精度不高的问题。同时，将传统协同过滤算法的预测已知评分与真实评分的差值作为特征，组合产生新的数据用于线性回归模型的训练，从整体上提高系统预测评分的准确性。下一步的工作将对新加入的用户特征和项目属性进行分析，以及如何发现并解决在训练过程中产生模型过拟合问题进行研究。

参考文献：

[1] Gouws R H, Tarp S. Information overload and data overload in lexicography [J]. International Journal of Lexicography, 2017: ecw030.

[2] Bouzid M, BonnefoyD, Lhuillier N, *et al.* Recommender system: US, WO//2010//005942 [P]. 2010.

[3] Patil V A, Ragha L. Comparing performance of collaborative filtering algorithms [C]// Proc of International Conference on Communication, Information & Computing Technology. 2012: 1-6.

[4] Al-Shamri M Y H. Power coefficient as a similarity measure for memory-based collaborative recommender systems [J]. Expert Systems with Applications, 2014, 41 (13): 5680-5688.

[5] 项亮. 推荐系统实践 [M]. 北京：人民邮电出版社, 2012.

[6] Ocepek U, Rugelj J, Bosnić Z. Improving matrix factorization recommendations for examples in cold start [J]. Expert Systems with Applications, 2015, 42 (19): 6784-6794.

[7] Vozalis M G, Margaritis K G. Applying SVD on item-based filtering [C]// Proc of International Conference on Intelligent Systems Design and Applications. [S. l.] : IEEE Computer Society, 2005: 464-469.

[8] Vozalis M G, Margaritis K G. Using SVD and demographic data for the enhancement of generalized collaborative filtering [J]. Information Sciences, 2007, 177 (15): 3017-3037.

[9] Gong S. A collaborative filtering recommendation algorithm based on user clustering and item clustering [J]. Journal of Software, 2010, 5 (7): 745-752.

[10] Li W, He W. An improved collaborative filtering approach based on user ranking and item clustering [M]// Internet and Distributed Computing Systems. 2013: 134-144.

[11] Zhang J, Lin Y, Lin M, *et al.* An effective collaborative filtering algorithm based on user preference clustering [J]. Applied Intelligence, 2016, 45 (2): 230-240.

[12] Pirlo G, Impedovo D. Cosine similarity for analysis and verification of static signatures [J]. Iet Biometrics, 2013, 2 (4): 151-158.

[13] He X, Luo Y. Mutual information based similarity measure for collaborative filtering [C]// Proc of IEEE International Conference on Progress in Informatics and Computing. 2010: 1117-1121.

[14] Alshamri M Y H, Alashwal N H. Fuzzy-weighted similarity measures for memory-based collaborative recommender systems [J]. Journal of Intelligent Learning Systems & Applications, 2014, 6 (1): 1-10.

[15] 陈震，谢峰，冯喜伟，等. 基于线性回归的推荐方法及系统, CN103942298A [P]. 2014.

[16] 王兆国，谢峰，关毅，等. 一种基于线性回归的新型推荐方法 [J]. 智能计算机与应用, 2017, 7 (4): 1-5.

[17] GroupLens [EB/OL]. http://www. grouplens. org/.